

Tips and FAQ QBO

Last Modified on 10/21/2021 5:48 am EDT

QuickBooks Online - How Can I Void A Bill-Payment?

The following API call allows voiding a **bill-payment** in QuickBooks Online:

```
curl -X PATCH \
https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/bill-payments/IdOfTheBillPayment/void \
-H 'Authorization: User *****, Organization *****, Connector *****' \
-H 'accept: application/json' \
```

After the bill-payment is voided, a *privateNote* with the *Voided* message appears in the response body of the **bill-payment**. Here is an example of the above API response:

```
{
  "checkPayment": {
    "printStatus": "NOT_SET",
    "bankAccountRef": {
      "name": "10100 Checking",
      "value": "58"
    }
  },
  "systemId": "8730",
  "syncToken": "1",
  "vendorRef": {
    "name": "Patton Hardware Supplies",
    "value": "11"
  },
  "metaData": {
    "createTime": "2019-06-17T05:54:39Z",
    "lastUpdateTime": "2019-06-17T06:19:30Z"
  },
  "payType": "CHECK",
  "totalAmt": 0,
  "privateNote": "Voided - RC 93",
  "sparse": false,
  "currencyRef": {
    "name": "United States Dollar",
    "value": "USD"
  },
  "exchangeRate": 1,
  "domain": "QBO",
  "id": "8730|1",
  "txnDate": "2019-06-16T00:00:00Z"
}
```

Quickbooks Online - Is it Possible to Fetch Custom Reports

Presently, fetching custom reports is not supported by the vendor API.

QuickBooks Online - Can I search for an invoice or bill by a customer's name

No, QuickBooks allows users to search for customers by 'customerRef.value'. This means that you would have to first make a GET/customers and search by the name. Then you can make your API call for what you are searching for.

QuickBooks Online - Can I use the connector for QuickBooks versions other than the US version

Yes. Currently QuickBooks Online supports development of apps in Australia, Canada, India, United Kingdom, and United States.

QuickBooks Online - Does GET /purchase-orders Support Query By Purchase Order

No, querying by pOStatus is not a supported feature of the QuickBooks Online API. You can perform a full selection from PurchaseOrder and then loop through the result set to pull out the open order details needed. A custom transformation or

formula filter is recommended because the QBO PurchaseOrder API does not support using the poststatus='OPEN' within the where clause.

Quickbooks Online - How Can I Add a New Custom Field via the API

Custom fields cannot be added via API calls. Quickbooks does not support the custom fields creation via API calls. This feature is only supported via the Quickbooks UI.

Quickbooks Online - How Can I Change the Status of a Payment

The status for a given payment cannot be changed. The status for a payment object is always set to PAID.

QuickBooks Online - How can I get my application approved for the QuickBooks App Store

There are currently no known blockers to getting approved on the QuickBooks App Store when utilizing SAP. A few of our users have been successful in getting their app approved for the QuickBooks App Store. More information on the approval process can be found on the [QuickBooks Developer Documentation](#).

Quickbooks Online - How Can I Permanently Delete a Vendor

The DELETE API call does not erase data, it performs only a SOFT DELETION in Quickbooks Online. Rather than remove the vendor, Quickbooks sets the field "inactive" to "true" when a delete API call is made. The soft deleted records can be later queried using this field.

Quickbooks Online - How Can I Query any QBO resource using OR or JOIN operators

Quickbooks Online has the following OCNQL limitations:

- OR operator is not supported in where clauses;
- GROUP BY and JOIN operator are not supported;
- Wildcard character support with LIKE clauses is limited to "%" (wildcard that substitutes for 0 or more characters);
- The response set returns all properties for each object (that is, projections are not supported).

Quickbooks Online - How Can I Query the GET/invoices Call Using the Purchase Order Number

Querying invoices according to the purchase order number is unavailable because the purchase order number associated with an invoice is the TxnId field, which is nested inside the entity LinkedTxn. In the Invoice object, the entity LinkedTxn is not queryable.

QuickBooks Online - How frequently does Quickbooks require an Oauth Refresh

QuickBooks Online requires a token refresh every 180 days. The refresh cannot occur any sooner than 150 days. We handle the oauth refresh for you in the background by maintaining the last refresh date and determining if a refresh is necessary on each API call.

QuickBooks Online - Production OAuth Key

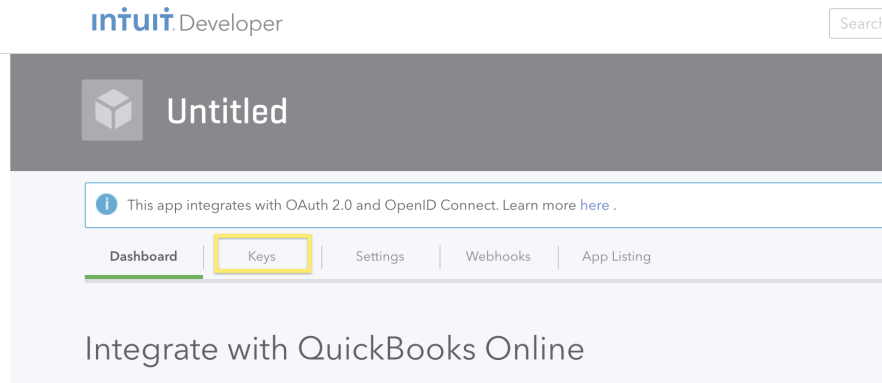
In QuickBooks, there is a difference between developer keys and production keys. If you create an OAuth2 connected app, chances are that app has developer keys. These keys will not allow you to connect to other QuickBooks companies. Reference this documentation for getting production ready keys: [Production Keys](#)

QuickBooks Online - QuickBooks Online Sandboxes

QuickBooks Online (QBO) offers free sandboxes to aid in the development of custom QBO applications. Sandbox application credentials can be used to authenticate QBO connector instances and perform all API calls that normal, non-sandbox applications are capable of. Follow these instructions to create your own QBO sandbox application:

1. Navigate to the Intuit Developer Portal: <https://developer.intuit.com/>. Sign up for a developer account if you do not already have one by clicking "Sign Up" in the upper-right corner.

- Navigate to <https://developer.intuit.com/v2/ui/#/app/startcreate>.
- Click the "Select APIs" button under "Just start coding".
- Select the APIs you wish to use. This selection can always be changed later via your app's settings.
- Click the "Keys" tab to be redirected to a page which contains your API key and secret. This is also where you configure settings for your applications such as which sandbox to connect to (one is automatically created for you when you sign up for a developer account).



- You should now be able to authenticate a QBO a connector instance using the credentials on the "Keys" tab. For more information on authenticating an instance of QuickBooks Online, see [this article](#).

Note: remember to include `connect.to.sandbox: true` in your POST body to `/instances` when authenticating an instance using a sandbox application.

Quickbooks Online - Reports and Filters

Different types of Quickbooks Online reports can be accessed through the GET `/reports/{id}` endpoint. Each report has their own supported filter fields that can be used to query. For example, if we're looking to retrieve the 'AgedPayableDetail' report that has a start date of 2019-01-01, an end date of 2019-03-21, and a report date of 2019-02-21, we'll supply the filter along with the value in our 'where' clause.

Curl Request example:

```
curl -X GET \
  "https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/reports/AgedPayableDetail?where=startDueDate%3D%272014-01-01%27%20AND%20endDueDate%3D%272019-12-21%27%20AND%20reportDate%3D%272019-03-21%27" \
  -H 'Authorization: User xxxxxxxx, Organization xxxxxxxx, Connector xxxxxxxx' \
  -H 'accept: application/json' \
```

Reports and Supported Filters

Name of Report	Supported Filter Fields
AccountList	startCreatedDate, endCreatedDate, createdDateRange, startUpdatedDate, endUpdatedDate, updatedDateRange, columns, accountStatus, orderBy, sortOrder, accountType
AgedPayables	reportDate, dateRange, agingPeriod, numberOfPeriods, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
AgedPayableDetail	startDueDate, endDueDate, reportDate, columns, term, pastDue, agingMethod, agingPeriod, numberOfPeriods, vendor
AgedReceivables	reportDate, dateRange, agingPeriod, numberOfPeriods, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
AgedReceivableDetail	startDueDate, endDueDate, reportDate, columns, term, pastDue, agingMethod, agingPeriod, numberOfPeriods, customer
BalanceSheet	startDate, endDate, dateRange, accountingMethod, groupBy, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
CashFlow	startDate, endDate, dateRange, groupBy, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy

ClassSales	startDate, endDate, dateRange, accountingMethod, groupBy, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
CustomerBalance	reportDate, dateRange, groupBy, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
CustomerBalanceDetail	startDueDate, endDueDate, agingMethod, reportDate, columns, orderBy, sortOrder, customerIds, vendorIds, itemIds, classIds, departmentIds
CustomerIncome	startDate, endDate, dateRange, accountingMethod, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
CustomerSales	startDate, endDate, dateRange, columns, accountingMethod, groupBy, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
DepartmentSales	startDate, endDate, dateRange, accountingMethod, groupBy, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
GeneralLedger	startDate, endDate, dateRange, columns, accountingMethod, accountStatus, orderBy, sortOrder, accountType, sourceAccountType, accountIds, sourceAccountIds, customerId, vendorIds, itemIds, classIds, departmentIds
ItemSales	startDate, endDate, dateRange, columns, accountingMethod, groupBy, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
InventoryValuationSummary	reportDate, dateRange, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
ProfitAndLoss	startDate, endDate, dateRange, accountingMethod, groupBy, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
ProfitAndLossDetail	startDate, endDate, dateRange, columns, accountingMethod, groupBy, orderBy, sortOrder, paymentMethod, accountType, customerIds, vendorIds, itemIds, classIds, departmentIds
TransactionList	vendorIds, transactionType, startDate, startCreatedDate, startUpdatedDate, startDueDate, endDate, endCreatedDate, endUpdatedDate, endDueDate, dateRange, createdDateRange, updatedDateRange, dueDateRange, sourceAccountType, sortOrder, qzurl, printed, paymentMethod, name, groupBy, memo, docnum, departmentIds, customer accountingMethod, apPaid, arPaid, classIds, cleared, bothamount
TrialBalance	startDate, endDate, dateRange, accountingMethod, summarizeBy
VendorBalance	reportDate, dateRange, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy
VendorBalanceDetail	startDueDate, endDueDate, reportDate, columns, dateRange, dueDateRange, orderBy, sortOrder, term, apPaid, agingMethod, vendor, department
VendorExpenses	startDate, endDate, dateRange, accountingMethod, customerIds, vendorIds, itemIds, classIds, departmentIds, summarizeBy

Filter definitions

- **dateRange** indicating the date range to use to filter the report. See your specific resource for valid values for this parameter: <https://developer.intuit.com/docs/api/accounting> Report resources > query parameters > date_macro
- **createdDateRange** indicating the date range to use to filter the report by create date.
- **updatedDateRange** indicating the date range to use to filter the report by updated date.

The supported values for date ranges are - *Today, Yesterday This Week, This Week-to-date, Last Week, Last Week-to-date, Next Week, Next 4 Weeks, This Month, This Month-to-date, Last Month, Last Month-to-date, Next Month, This Fiscal Quarter, This Fiscal Quarter-to-date, Last Fiscal Quarter, Last Fiscal Quarter-to-date, Next Fiscal Quarter, This Fiscal Year, This Fiscal Year-to-date, Last Fiscal Year, Last Fiscal, Year-to-date, Next Fiscal Year.* The default is This Fiscal Year-to-date.

- **startDate** indicating the start date of a period for the report. The date format is 'YYYY-MM-DD', e.g., '2015-10-03'
- **endDate** indicating the end date of a period for the report. The date format is 'YYYY-MM-DD', e.g., '2015-10-03'
- **startDueDate** indicating the start date of a period for the report. The date format is 'YYYY-MM-DD', e.g., '2015-10-03'
- **endDueDate** indicating the end date of a period for the report. The date format is 'YYYY-MM-DD', e.g., '2015-10-03'

- **startCreatedDate** indicating the start created date of a period for the report. The date format is 'YYYY-MM-DD', e.g., '2015-10-03'
- **endCreatedDate** indicating the end created date of a period for the report. The date format is 'YYYY-MM-DD', e.g., '2015-10-03'
- **startUpdatedDate** indicating the start created date of a period for the report. The date format is 'YYYY-MM-DD', e.g., '2015-10-03'
- **endUpdatedDate** indicating the end created date of a period for the report. The date format is 'YYYY-MM-DD', e.g., '2015-10-03'
- **reportDate** provided with the date format of *YYYY-MM-DD*. The default value is today's date.
- **accountingMethod** with supported values of Cash or Accrual. The default is that defined in the company preferences in Quickbooks.
- **accountType** indicating the account type to use for the filter.
- **sourceAccountType** indicating the source account type to use for the filter.

The supported values for account types, including the source account type, are - Bank, AccountsReceivable, OtherCurrentAsset, FixedAsset, OtherAsset, AccountsPayable, CreditCard, OtherCurrentLiability, LongTermLiability, Equity, Income, CostOfGoodsSold, Expense, OtherIncome, OtherExpense, NonPosting. The default is to include all account types in the report.

- **agingMethod** with supported values of Report_date or Current. The default is Current.
- **agingPeriod** indicating the number of days in the aging period. The default is 30.
- **numberOfPeriods** indicating the number of periods to be shown in the report. The default is 4.
- **arPaid** with supported values of All, Paid, Unpaid. The default is Unpaid.
- **apPaid** with supported values of All, Paid, Unpaid. The default is Unpaid.
- **pastDue** indicating to number of past due days to use for the filter.
- **shipVia** indicating the name of the shipping method as specified in the invoice.
- **paymentMethod** indicating the name of the payment method used, with supported values of Cash, Check, Dinners Club, American Express, Discover, MasterCard, Visa.
- **columns** a list of comma separated column names in an 'in ()' clause, indicating the column types to be shown in the report.
- **orderBy** which specifies the column for the sort order for the results. The default is tx_date.
- **sortOrder** which specifies the sort order for the results. The supported values are 'ascend' and 'descend'. The default is ascend.
- **groupBy** which specifies a criteria to group the the results. The supported values are *Total, Month, Week, Days, Quarter, Year, Customers, Vendors, Classes, Departments, Employees, ProductsAndServices*. The default value is Total.
- **accountIds** A comma separated list of account IDs to filter by.
- **sourceAccountIds** A comma separated list of the source account IDs to filter by.
- **customerIds** A comma separated list of customer IDs to filter by.
- **vendorIds** A comma separated list of vendor IDs to filter by.
- **itemIds** A comma separated list of item IDs to filter by.
- **classIds** A comma separated list of class IDs to filter by.
- **termIds** A comma separated list of term IDs to filter by.
- **departmentIds** A comma separated list of department IDs to filter by.

Quickbooks Online - Uploading Invoice Attachments

Uploading attachments to specific objects in Quickbooks Online (QBO) including Invoices and Bills is a straight-forward process, but if you are using the connector API Docs to upload these attachments there is one important note to make. The /attachments endpoints for these objects are available towards the bottom of the API docs within this resource:

```
/{objectName}/{id}/attachments
```

The reason for this is to keep the API docs more in line with how QBO actually handles the [upload of attachments to objects](#) as a separate API and not as a function of the Transaction Resource APIs themselves. For example you can make POST calls with the uploaded files to these endpoints:

```
/invoices/{id}/attachments
/bills/{id}/attachments
```

Quickbooks Online - What are the Available Values for a Vendor's BillableStatus

A Vendor's "**BillableStatus**" can have the following values: *Billable, NotBillable* and *HasBeenBill*.

QuickBooks Online - Why am I seeing "Authentication with the provider failed"

If you are seeing the below error response when attempting to perform a QuickBooks Online call:

```
{ "requestId": "xxxxxxxxxx",  
  "message": "Authentication with the provider failed. OAuth token refresh may have encountered problems. If problem continues, edit your connector instance and ensure that access has not been revoked from the provider.",  
  "providerMessage": "3200 - message=ApplicationAuthenticationFailed; errorCode=003200; statusCode=401 - null" }
```

You can test a shorter refresh interval by using Intuit's Developer Playground using the below steps:

1. Go to <https://appcenter.intuit.com/Playground/OAuth/IA>. Populate the key and secret and then specify how much time (in seconds) you would like to test for the expiration of your tokens.
2. Click "Connect to Quickbooks". Go through the OAuth flow.
3. After you have completed the OAuth flow, there will be a page that includes a realm id, token and secret. You will use these to POST <https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/instances>. You will use the following payload:

```
{  
  "element": {  
    "key": "quickbooks"  
  },  
  "configuration": {  
    "oauth.callback.url": "http://www.cloud-elements.com",  
    "oauth.user.refresh_interval": "",  
    "quickbooks.realm.id": "",  
    "oauth.user.token": "",  
    "oauth.user.token.secret": "",  
    "oauth.api.key": "xxxxxxxxxxxxxxxxxxxx",  
    "oauth.api.secret": "xxxxxxxxxxxxxxxx",  
    "quickbooks.datasource": "QBO",  
    "oauth.user.refresh_time": "http://www.epochconverter.com/>"  
  },  
  "tags": [  
    "QBO Token"  
  ],  
  "name": "QBO Token",  
  "externalAuthentication": "initial"  
}
```

4. After you have posted the instance, continually call one of the endpoints from the instance until the refresh interval has expired. For example, if you set it to 120 seconds, make sure you are making API calls for at least two minutes.
5. Immediately after the refresh interval has passed, do another call to ensure it is still working.
6. Do a GET /instances/{id}. Review the oauth.user.refresh_time and see it has been updated to when the token was refreshed.

NOTE: due to the shorter time interval, you run the risk of getting a "Refresh out of bounds" error message if you don't attempt it within a specified time frame. This is a QuickBooks limitation.

QuickBooks Online - Why do I get "INTERNAL_SERVER_ERROR" when syncing customer invoices

The below error might be seen when trying to sync customer invoices:

```
java.lang.RuntimeException: {"message":"getCustomerById failure","endpoint":"https://console.cloud-elements.com/elements/api-v2/hubs/finance/customers/1042","headers":{"Content-Type":["application/json"],"Authorization":["Element xxxxxxxxxxxxxxxx=, User=xxxxxxxxxxxxxxxx"]},"statusCode":"INTERNAL_SERVER_ERROR","responseBody":{"requestId":"xxxxxxxxxxxx","message":"Unknown internal error"}}
```

QuickBooks Online only allows one connection per user ID. If you create one instance, and a second instance with the same user ID, then the second instance will revoke access to the first one.

QuickBooks online has a sync token attached to each ID. When you perform a GET /invoices call, the ID will appear as: 1234|3. This translates to |

The sync token helps manage concurrent requests. Only the request with the most recent sync token will succeed. When you PATCH /invoices, send the ID with the most recent sync token.

QuickBooks Online - Why do I see "This app is not set up to allow connection from your country" when creating an instance

This error is seen when you attempt to create an instance from a country that is not enabled on the QuickBooks Online app. QuickBooks Online allows you to choose several countries for each app. Add the country where you are attempting to connect from as explained in [QuickBooks Online Documentation](#).

Quickbooks Online - OAuth1 Migration Script

At the end of 2019, Quickbooks Online is deprecating their OAuth1 service in favor of using OAuth2. After **December 17th, 2019**, Quickbooks will be revoking all OAuth1 tokens and no new tokens will be granted. To avoid connection failures, you will need to migrate your OAuth1 QBO connector instances to OAuth2.

SAP Open Connectors is happy to provide the following migration scripts that will handle this for you at this [Github Link](#). Essentially, there are two scripts:

- the first to find all the OAuth1 QBO instances that need to be migrated
- the second to migrate those instances to OAuth2

You will need to supply the following pieces of information to the first script:

- User Secret
- Organization Secret
- Your SAP Open Connectors Environment (Staging or Prod)

You will need to supply the previous information into the second script, as well as:

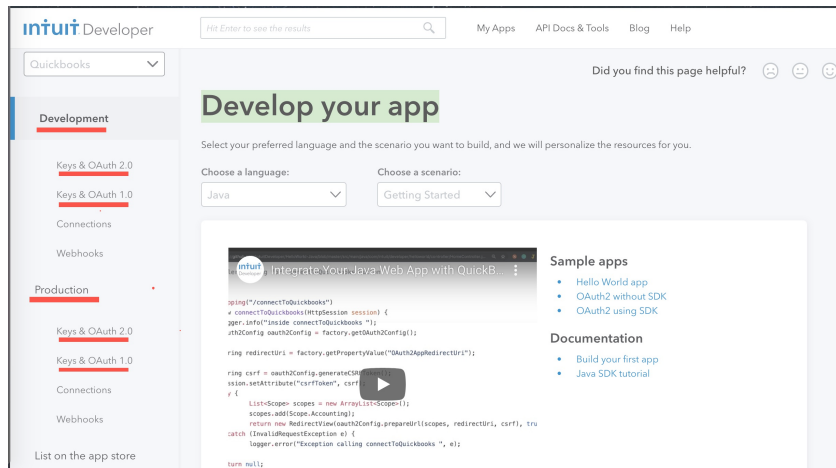
- OAuth 2.0 Client Id associated with your OAuth 1.0 application
- OAuth 2.0 Client Secret associated with your OAuth 1.0 application
- List of instances procured by running the first script

Special Notes:

- The migration endpoint is /migrate-tokens
- In your header, you still need your full Auth Header which includes User+Organization+Element.
- In the payload, you want to include the following:

```
{
  "client_id": "XXXXX",
  "client_secret": "XXXXX"
}
```

- Ensure you are using the proper key type, Development vs Production. You want to use the matching key types based on your QBO app. So if the OAuth1 app instance is using Production keys you will want to use OAuth2 Production keys for the migration.
- Note the areas underlined in Red to find Keys



- The `oauth.callback_url` on the existing OAuth1 instance must match the Redirect URIs setup in the OAuth2 app.
- Quickbooks only accepts redirect urls that use HTTPS and does not support HTTP, therefore you will need to make sure the redirect url's in your QBO app are using HTTPS for any instance you want to migrate.
- Only valid OAuth1 tokens can be migrated to OAuth2 so if you have an invalidated instance returning 401s it must be re-authenticated prior to OAuth2 migration.